

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re Application of

Louis R. Degenaro, et al.

Serial No.: 10/665,564

Group Art Unit: 2165

Filed: September 22, 2003

Examiner: Syed, Farhan M.

For: VIRTUAL RESOURCES METHOD, SYSTEM, AND SERVICE

Honorable Commissioner of Patents
Alexandria, VA 22313-1450
MAIL STOP Appeal Brief-Patents

APPELLANTS' BRIEF ON APPEAL UNDER 35 U.S.C. §134(a)

Sir:

Appellants respectfully appeal the decision of the Examiner in the final rejection of claims 1-37 in the Final Office Action mailed November 3, 2010.

I. REAL PARTY OF INTEREST

The real party of interest is International Business Machines Corporation, assignee of 100% interest of the above-referenced patent application, as recorded on September 22, 2003 with the United States Patent and Trademark Office on reel/frame 014530/0523.

II. RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences known to Appellants,

Appellants' legal representative or Assignee, which would directly affect or be directly affected by or have a bearing on the Board's decision on this appeal.

III. STATUS OF CLAIMS

Claims 1-37, all the claims presently pending in the application, stand rejected under 35 U.S.C. § 102(e) as being allegedly anticipated by Severin (U.S. Patent Pub. No. 2005/0005261). Appellants respectfully appeals this rejection.

IV. STATUS OF AMENDMENTS (SUBSEQUENT TO FINAL REJECTION)

A request for reconsideration under 35 USC §1.116 filed on January 3, 2011 after final rejection with no claim amendments, was considered by the Examiner. In an advisory action dated January 18, 2011, the Examiner indicated that the request for reconsideration was considered but did not place the application in condition for allowance.

Therefore, the attached claims reflect the version of the claims in the amendment under 37 CFR §1.111 filed on August 31, 2010.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The bases in the specification for the claims are as follows:

1. (Rejected) A method of developing actual resources without alteration {e.g., line 21 of page 11 through line 5 of page 12; line 20 of page 12 through line 1 of page 13; lines 20-16 of page 36} into a collection of virtual resources customized to a particular audience {e.g., line 13 of page 1 through line 22 of page 2; lines 1-7 of page 32}, said method comprising:

constructing at least one virtual resource independent of an actual resource {e.g., line 21 of page 14 through line 8 of page 15; line 21 of page 15 through line 2 of page 16; lines 3- 6 and 23-24 of page 18; 1240 of Figure 12; lines 20-23 of page 30};

connecting the actual resource to the at least one virtual resource {e.g., lines 3-6 of page 18; lines 19-21 of page 19; ref. no. 1240 of Figure 12; lines 20-23 of page 30};

retrieving the at least one virtual resource from the tangible computer readable media {e.g., lines 7-16 of page 18; Figure 5; lines 6-8 of page 21; ref. no. 1250 of Figure 12; lines 22-23 of page 30}; and

extracting at least one descriptor from said at least one retrieved virtual resource {e.g.,Figure 5; line 15 of page 21 through line 2 of page 22},

wherein said virtual resource comprises a resource utilized at a logic authoring time and said actual resource comprises a resource utilized at a runtime {e.g., lines 6-8 of page 15}.

9. (Rejected) The method of claim 1,
wherein information constructing the at least one virtual resource includes
data independent from the actual resource {e.g., **line 14 of page 12 through line**
5 of page 25}, the method further comprising:
selectively manipulating the retrieved virtual resource by updating or
deleting at least a portion of the retrieved virtual resource{e.g., **line 20 of page**
17 through line 2 of page 18}; and
authoring the virtual resource into a logic code stored and executable by
the computer to generate a second actual resource from the virtual resource {e.g.,
lines 17-24 of page 13 }.

14. (Rejected) A system for developing actual resources without alteration
{e.g., **line 21 of page 11 through line 5 of page 12; line 20 of page 12 through**
line 1 of page 13; lines 20-16 of page 36} into a collection of virtual resources
customized to a particular audience {e.g., e.g., **line 13 of page 1 through line 22**
of page 2; lines 1-7 of page 32}, said system comprising:

means for constructing at least one virtual resource independent of at least
one actual resource, using a processor on a computer {e.g., **line 21 of page 14**
through line 8 of page 15; line 21 of page 15 through line 2 of page 16; lines

3-6 and 23-24 of page 18; 1240 of Figure 12; lines 20-23 of page 30};

means for connecting at least one actual resource to at least one virtual resource {e.g., **lines 3-6 of page 18; lines 19-21 of page 19; 1240 of Figure 12; lines 20-23 of page 30};**

means for retrieving said at least one virtual resource {e.g., **lines 7-16 of page 18; Figure 5; lines 6-8 of page 21; 1250 of Figure 12; lines 22-23 of page 30}; and**

means for extracting at least one descriptor from said at least one retrieved virtual resource {e.g., **Figure 5; line 15 of page 21 through line 2 of page 22}, wherein said virtual resource comprises a resource utilized at a logic authoring time and said actual resource comprises a resource utilized at a runtime {e.g., lines 6-8 of page 15}.**

27. (Rejected) A service to manage descriptions of actual resources in a system comprised of a plurality of actual resources {e.g., **ref. no. 1210 of Fig. 12}, said service comprising:**

defining at least one virtual domain to satisfy a user-requirements analysis {e.g., **lines 5- 7 of page 32}; and**

defining, by a processor, at least one virtual resource describing at least one actual resource within the at least one virtual domain to satisfy the user-

requirements analysis {e.g., **lines 5-7 of page 32**},

wherein said virtual resource comprises a resource utilized at a logic authoring time and said actual resource comprises a resource utilized at a runtime {e.g., **lines 6-8 of page 15**},

wherein said virtual resource is stored on a tangible computer readable media accessible by the processor {e.g., **Figure 13, line 6 of page 38 through line17 of page 39**},

wherein said defining the at least one virtual resource relates in part to the actual resource and in part independent of the actual resource {e.g., **line 14 of page 12 through line 5 of page 25**}, and

wherein said defining the at least one virtual resource comprises creating and then manipulating the virtual resource at least in part independent of the actual resource {e.g., **line 14 of page 12 through line 5 of page 25**}.

31. (Rejected) A method of deploying computing infrastructure in which computer readable code is integrated into a computing system, such that said code and said computing system combine to perform a method of developing said actual resources without alteration {e.g., **line 21 of page 11 through line 5 of page 12; line 20 of page 12 through line 1 of page 13; lines 20-16 of page 36**} into a collection of virtual resources customized to a particular audience {e.g.,

line 13 of page 1 through line 22 of page 2; lines 1-7 of page 32}, said method comprising:

constructing at least one virtual resource independent of said actual resources {e.g., **line 21 of page 14 through line 8 of page 15; line 21 of page 15 through line 2 of page 16; lines 3-6 and 23-24 of page 18; 1240 of Figure 12; lines 20-23 of page 30};**

connecting at least one actual resource to at least one virtual resource {e.g., **lines 3-6 of page 18; lines 19-21 of page 19; 1240 of Figure 12; lines 20-23 of page 30};**

performing at least one retrieval of a virtual resource {e.g., **lines 7-16 of page 18; Figure 5; lines 6-8 of page 21; 1250 of Figure 12; lines 22-23 of page 30}; and**

extracting at least one descriptor from said at least one retrieved virtual resource {e.g., **Figure 5; line 15 of page 21 through line 2 of page 22},**

wherein said virtual resource comprises a resource utilized at a logic authoring time and said actual resource comprises a resource utilized at a runtime {e.g., **lines 6-8 of page 15}.**

32. (Rejected) A tangible computer-readable storage medium tangibly embodying a program of machine-readable instructions executable by a digital processing

apparatus to perform a method of developing said actual resources without alteration {e.g., **line 21 of page 11 through line 5 of page 12; line 20 of page 12 through line 1 of page 13; lines 20-16 of page 36**} into a collection of virtual resources customized to a particular audience {e.g., **e.g., line 13 of page 1 through line 22 of page 2; lines 1-7 of page 32**}, said method comprising:

constructing at least one virtual resource independent of said actual resources {e.g., **line 21 of page 14 through line 8 of page 15; line 21 of page 15 through line 2 of page 16; lines 3-6 and 23-24 of page 18; 1240 of Figure 12; lines 20-23 of page 30**};

connecting at least one of said actual resources to said at least one virtual resource {e.g., **lines 3-6 of page 18; lines 19-21 of page 19; 1240 of Figure 12; lines 20-23 of page 30**};

performing at least one retrieval of said virtual resource {e.g., **lines 7-16 of page 18; Figure 5; lines 6-8 of page 21; 1250 of Figure 12; lines 22-23 of page 30**}; and

extracting at least one descriptor from said at least one retrieved virtual resource {e.g., **Figure 5; line 15 of page 21 through line 2 of page 22**}, wherein said virtual resource comprises a resource utilized at a logic authoring time and said actual resource comprises a resource utilized at a runtime {e.g., **lines 6-8 of page 15**}.

33. (Rejected) A method of developing actual resources without alteration {e.g., **line 21 of page 11 through line 5 of page 12; line 20 of page 12 through line 1 of page 13; lines 20-16 of page 36**} into a collection of virtual resources customized to a particular audience {e.g., **line 13 of page 1 through line 22 of page 2; lines 1-7 of page 32**}, said method comprising:

constructing at least one virtual resource independent of an actual resource {e.g., **line 21 of page 14 through line 8 of page 15; line 21 of page 15 through line 2 of page 16; lines 3- 6 and 23-24 of page 18; 1240 of Figure 12; lines 20-23 of page 30**}; and

providing in the at least one virtual resource a structured meta-data layer which contains semantic information for leveraging by a consumer of the virtual resources {e.g., **line 2 of page 32 through line 5 of page 38**},

wherein said virtual resource comprises a resource utilized at a logic authoring time and said actual resource comprises a resource utilized at a runtime {e.g., **lines 6-8 of page 15**}.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Appellants present the following issue for review by the Board of Patent Appeals and Interferences:

Whether Claims 1-37 are not patentable under 35 U.S.C. § 102(e) over Severin (U.S. Patent Pub. No. 2005/0005261, hereinafter Severin '261). , as alleged by the Examiner.

VII. ARGUMENT

A. Rejection of Claims 1-37 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Severin (U.S. Patent Pub. No. 2005/0005261, hereinafter Severin '261).

1. Appellants' arguments with respect to Independent Claims 1, 14, 27, 31, 32 and 33.

The Examiner alleges that Severin '261 teaches claims 1-37. However, Severin '261 fails to teach a *method of developing actual resources without alteration into a collection of virtual resources customized to a particular audience, said method comprising: constructing at least one virtual resource independent of an actual resource; storing the virtual resource in a tangible computer readable media, using a processor on a computer; connecting the actual resource to the at least one virtual resource; retrieving the at least one virtual resource from the tangible computer readable media; and extracting at least one descriptor from said at least one retrieved virtual resource, wherein said virtual resource comprises a resource utilized at a logic authoring time, while said actual resource comprises a resource utilized at a runtime.*

Appellants previously noted that it also does not appear that the disclosure in the publication document of Severin '261 is fully supported by the provisional application (hereinafter Severin '251). Moreover, it is also noted that Severin '261 (non-provisional application) cannot be used as a reference since it has a later U.S. filing date than the present Application. Rather, the Examiner must use Severin '251, the provisional application only in order to argue a reasoning for the anticipation rejection.

Therefore, the disclosure of the non-provisional application Severin '261 should not even be referenced when comparing with the claimed invention since Severin '261 is after the filing date of the present application.

However, in the present Office Action, dated November 3, 2010, the Examiner argues that the Examiner finds support for construction in the provisional application, see page 6, which defines a constructor as a set of programming logic associated with the creation of an instance.)(paragraph [0248]) at least one virtual resource independent (a virtual model implementation. In addition, the Examiner finds support for a virtual model implementation as a model controller on page 24 of the provisional application.)(paragraph [0248]) of an actual resource (i.e. real implementation. The Examiner further alleges that he finds support for actual resource in the view controller of the provisional application, see page 25.)(paragraph [0248; 0550]).

However, the constructor is provided only as a definition with no connection with the remaining portion cited in page 24 and 25 of the provisional application.

Moreover, on page 24 of the provisional application, it merely recites that it "reduces direct object connections (by connecting objects through event handling) and improves code reuse." However, connections are still shown in the drawings even if they are not direct. Moreover, the disclosure must be enabling and not in reference to the later non-provisional application which should not be used. The non-provisional application cannot be used at all as mentioned above.

In addition, page 20 recites that a CI engine creates connections between objects dynamically at run-time instead of creating object connections at compile-time which is referring to the connections and not the resources. Moreover, on page 21, the provisional application is referring to a teaching that it is not obvious to use a high-level language to write only the parts of a system, then use builders and descriptions of parts to assemble a complex computer system rather than the reference to the run-time.

The Examiner uses meta data from paragraph [0049] as a descriptor from the virtual resource, however, the metadata used in the provisional application is not limited to being utilized at logic authoring time as claimed.

In fact the provisional application Severin '251 on page 9 recites that " It is an object of this invention to allow for the full description of any structured data

(including objects) at run-time. This mechanism will be referred to as the "Metadata Design Pattern" and has not been recognized outside of this invention. Therefore, not only does Severin '251 fail to teach (or even suggest) such a limitation of extracting at least one descriptor from said at least one retrieved virtual resource, wherein said virtual resource comprises a resource utilized at a logic authoring time, while said actual resource comprises a resource utilized at a runtime, but in fact Severin '251 teaches away from the claimed invention.

Therefore, respectfully, there is also no enabling link provided between the different parts of cited provisional application such that there is a teaching or suggestion of the entire claim of constructing at least one virtual resource independent of an actual resource; storing the virtual resource in a tangible computer readable media, using a processor on a computer; connecting the actual resource to the at least one virtual resource; retrieving the at least one virtual resource from the tangible computer readable media; and extracting at least one descriptor from said at least one retrieved virtual resource, wherein said virtual resource comprises a resource utilized at a logic authoring time, while said actual resource comprises a resource utilized at a runtime.

In addition as noted previously, according to MPEP §2131, the following applies in a 35 U.S.C. §102 rejection:

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a

single prior art reference." Verdegaal Bros. v. Union Oil Co. of California, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).. "The identical invention must be shown in as complete detail as is contained in the ... claim." Richardson v. Suzuki Motor Co., 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). The elements must be arranged as required by the claim, but this is not an ipsissimis verbis test, i.e., identity of terminology is not required. In re Bond, 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990).

Therefore, as shown above, since the provisional application of Severin '251, alone and without the benefit of the disclosure of the non-provisional application Severin '261, does not identically disclose the invention in as complete detail as contained in the claim, the anticipation rejection must be removed.

The other independent claims are not anticipated by Severin '251 in a similar manner to claim 1.

Therefore, Appellants submit that there are elements of the claimed invention that are not taught or suggest by the provisional application of Severin '251. Severin '261, as mentioned above cannot be used in the rejection in any manner or to explain the provisional application since it dates after the filing date of the present Application.

Therefore, Appellants again submit that there clearly are elements of the claimed invention that are not taught by Severin '261, and the Board is respectfully requested to reconsider and withdraw this rejection.

VIII. CONCLUSION

In view of the foregoing, Appellants submit that claims 1-37, all of the claims presently pending in the application, are patentably distinct over the prior art of record and are in condition for allowance. Thus, the Board is respectfully requested to remove the rejections of claims 1-37.

The Commissioner is hereby authorized to charge any deficiency in fees or to credit any overpayment in fees to Assignee's Deposit Account No. 50-0510.

Respectfully Submitted,



Date: April 4, 2011

Sam S. Sahota, Esq.
Registration No. 47,051

Sean M. McGinn, Esq.
Registration No. 34,386

**MCGINN INTELLECTUAL PROPERTY
LAW GROUP, PLLC**
8321 Old Courthouse Road, Suite 200
Vienna, VA 22182-3817
(703) 761-4100
Customer No. 48150

IX. CLAIMS APPENDIX

1. (Rejected) A method of developing actual resources without alteration into a collection of virtual resources customized to a particular audience, said method comprising:

constructing at least one virtual resource independent of an actual resource; storing the virtual resource in a tangible computer readable media, using a processor on a computer;

connecting the actual resource to the at least one virtual resource; retrieving the at least one virtual resource from the tangible computer readable media; and

extracting at least one descriptor from said at least one retrieved virtual resource,

wherein said virtual resource comprises a resource utilized at a logic authoring time, while said actual resource comprises a resource utilized at a runtime.

2. (Rejected) The method of claim 1, wherein said connecting comprises directly mapping the at least one actual resource to the at least one virtual resource, and

wherein the constructing the at least one virtual resource comprises including constraints upon the virtual resources without altering the actual resources.

3. (Rejected) The method of claim 1, wherein the constructing comprises at least one of:

- renaming a method;
- hiding a method;
- composing a method;
- renaming an attribute;
- hiding an attribute;
- composing an attribute;
- assigning to at least one domain;
- designating as a collection;
- assigning to at least one validator;
- assigning a description;
- designating as at least one of ready and not ready; and
- assigning a last modified date and time.

4. (Rejected) The method of claim 1, wherein said at least one virtual resource

comprises a plurality of virtual resources and said virtual resources are connected to each other through a relationship carrying semantic that can be leveraged by a consumer of resources, said method further comprising:

constructing at least one virtual relationship between at least two virtual resources;

coupling at least one actual relationship implementor to at least one virtual relationship;

performing at least one retrieval of a virtual relationship; and

extracting at least one descriptor from at least one retrieved virtual relationship.

5. (Rejected) The method of claim 4, wherein said coupling comprises:

directly mapping said at least one actual relationship implementor to said at least one virtual relationship.

6. (Rejected) The method of claim 4, wherein the relationship constructing comprises at least one of:

assigning a root virtual resource name;

assigning a target virtual resource name;

assigning a relationship name;

assigning a relationship type;
assigning a description;
assigning a target instance naming scheme;
designating as at least one of ready and not ready; and
assigning a last modified date and time.

7. (Rejected) The method of claim 4, wherein the retrieving comprises locating virtual

relationships by at least one of:

a domain;
a name;
a root;
a type; and
a target.

8. (Rejected) The method of claim 1, wherein virtual resources are connected to each other, said method further comprising:

constructing at least one virtual relationship between at least two virtual resources;

coupling at least one actual relationship implementor to at least one virtual

relationship;

performing at least one retrieval of a virtual relationship; and

extracting at least one descriptor from at least one retrieved virtual relationship,

wherein said coupling comprises directly mapping said at least one actual relationship implementor to said at least one virtual relationship.

9. (Rejected) The method of claim 1, wherein information constructing the at least one virtual resource includes data independent from the actual resource, the method further comprising:

selectively manipulating the retrieved virtual resource by updating or deleting at least a portion of the retrieved virtual resource; and

authoring the virtual resource into a logic code stored and executable by the computer.

10. (Rejected) The method of claim 8, wherein the relationship constructing comprises at least one of:

assigning a root virtual resource name; assigning a target virtual resource name; assigning a relationship name;

assigning a relationship type;
assigning a description;
assigning a target instance naming scheme;
designating as at least one of ready and not ready; and
assigning a last modified date and time.

11. (Rejected) The method of claim 1, wherein the retrieving comprises locating virtual resources by at least one of:

a domain;
a name; and
a relationship.

12. (Rejected) The method of claim 8, wherein the retrieving comprises locating virtual relationships by at least one of:

a domain;
a name;
a root;
a type; and
a target.

13. (Rejected) The method of claim 2, wherein descriptor validator information is employed to limit actual resource usage.

14. (Rejected) A system for developing actual resources without alteration into a collection of virtual resources customized to a particular audience, said system comprising: means for constructing at least one virtual resource independent of at least one actual resource, using a processor on a computer;

means for connecting at least one actual resource to at least one virtual resource;

means for retrieving said at least one virtual resource; and

means for extracting at least one descriptor from said at least one retrieved virtual

resource,

wherein said virtual resource comprises a resource utilized at a logic authoring time while said actual resource comprises a resource utilized at a runtime.

15. (Rejected) The system of claim 14, wherein said means for connecting comprises means for directly mapping the at least one actual resource to the at least one virtual resource.

16. (Rejected) The system of claim 14, wherein the means for constructing performs at least one of:

- renaming a method;
- hiding a method;
- composing a method;
- renaming an attribute;
- hiding an attribute;
- composing an attribute;
- assigning to at least one domain;
- designating as a collection;
- assigning to at least one validator;
- assigning a description;
- designating as at least one of ready and not ready; and
- assigning a last modified date and time.

17. (Rejected) The system of claim 14, wherein virtual resources are connected to each other through a relationship carrying semantic that can be leveraged by a consumer of resources, comprising:

means for constructing at least one virtual relationship between at least two

virtual

resources;

means for coupling at least one actual relationship implementor to at least one virtual

relationship;

means for performing at least one retrieval of a virtual relationship; and

means for extracting at least one descriptor from at least one retrieved virtual relationship.

18. (Rejected) The system of claim 17, wherein said means for coupling comprises:

means for directly mapping said at least one actual relationship implementor to said at

least one virtual relationship.

19. (Rejected) The system of claim 17, wherein the means for constructing at least

one virtual relationship performs at least one of:

assigning a root virtual resource name;

assigning a target virtual resource name;
assigning a relationship name;
assigning a relationship type;
assigning a description;
assigning a target instance naming scheme;
designating as at least one of ready and not ready; and
assigning a last modified date and time.

20. (Rejected) The system of claim 14, wherein the means for retrieving performs locating virtual relationships by at least one of:

a domain;

a name;

a root;

a type; and

a target.

21. (Rejected) The system of claim 14, wherein virtual resources are connected to each other, said system further comprising:

means for constructing at least one virtual relationship between at least two virtual

means for coupling at least one actual relationship implementor to at least one virtual relationship;

means for performing as least one retrieval of a virtual relationship; and

means for extracting at least one descriptor from at least one retrieved virtual relationship.

22. (Rejected) The system of claim 21, wherein said means for coupling comprises: means for directly mapping said at least one actual relationship implementor to said at least one virtual relationship.

23. (Rejected) The system of claim 21, wherein the means for constructing a relationship comprises at least one of:

means for assigning a root virtual resource name;

means for assigning a target virtual resource name;

means for assigning a relationship name;

means for assigning a relationship type;

means for assigning a description;

means for assigning a target instance naming scheme;

means for designating as at least one of ready and not ready; and

means for assigning a last modified date and time.

24. (Rejected) The system of claim 21, wherein the means for retrieving comprises locating

virtual resources by at least one of:

a domain;

a name; and

a relationship.

25. (Rejected) The system of claim 21, wherein the means for retrieving comprises locating

virtual relationships by at least one of:

a domain;

a name;

a root; and

a target.

26. (Rejected) The system of claim 15, wherein descriptor validator information is

27. (Rejected) A service to manage descriptions of actual resources in a system comprised of a plurality of actual resources, said service comprising: defining at least one virtual domain to satisfy a user-requirements analysis; and defining, by a processor, at least one virtual resource describing as least one actual resource within the at least one virtual domain to satisfy the user-requirements analysis, wherein said virtual resource comprises a resource utilized at a logic authoring time while said actual resource comprises a resource utilized at a runtime, wherein said virtual resource is stored on a tangible computer readable media accessible by the processor, wherein said defining the at least one virtual resource relates in part to the actual resource and in part independent of the actual resource, and wherein said defining the at least one virtual resource comprises creating and then manipulating the virtual resource at least in part independent of the actual resource.

28. (Rejected) The service of claim 27, further comprising:

analyzing a requirement for actual resource usage, to provide said user requirements analysis.

29. (Rejected) The service of claim 27, further comprising:

defining at least one virtual relationship between at least two virtual resources.

30. (Rejected) The service of claim 29, wherein at least one of a virtual resource and a virtual relationship is utilized to create an application program.

31. (Rejected) A method of deploying computing infrastructure in which computer readable code is integrated into a computing system, such that said code and said computing system combine to perform a method of developing said actual resources without alteration into a collection of virtual resources customized to a particular audience, said method comprising:

constructing at least one virtual resource independent of said actual resources;

connecting at least one actual resource to at least one virtual resource;

performing at least one retrieval of a virtual resource; and

extracting at least one descriptor from said at least one retrieved virtual resource,

wherein said virtual resource comprises a resource utilized at a logic authoring time and said actual resource comprises a resource utilized at a runtime.

32. (Rejected) A tangible computer-readable storage medium tangibly embodying a program of machine-readable instructions executable by a digital processing apparatus to perform a method of developing said actual resources without alteration into a collection of virtual resources customized to a particular audience, said method comprising:

constructing at least one virtual resource independent of said actual resources;

connecting at least one of said actual resources to said at least one virtual resource;

performing at least one retrieval of said virtual resource; and

extracting at least one descriptor from said at least one retrieved virtual resource,

wherein said virtual resource comprises a resource utilized at a logic authoring time and said actual resource comprises a resource utilized at a runtime.

33. (Rejected) A method of developing actual resources without alteration into a collection of virtual resources customized to a particular audience, said method comprising: constructing at least one virtual resource independent of an actual resource; and

providing in the at least one virtual resource a structured meta-data layer which contains semantic information for leveraging by a consumer of the virtual resources,

wherein said virtual resource comprises a resource utilized at a logic authoring time while said actual resource comprises a resource utilized at a runtime.

34. (Rejected) The method of claim 33, wherein said semantic information includes relationships with agreed upon semantics including any of "related-to", "contains", and "is-conflicting-with", between entities.

35. (Rejected) The method of claim 33, wherein said semantic information allows any of making new resource manipulation operations available to logic authoring tools and serving as an input to a conflict detection tool, and wherein a common layer is provided to resource utilizing tools comprising the logic authoring tools to directly access the virtual resource.

36. (Rejected) The method of claim 1, further comprising:
creating at least one virtual resource instance;
assigning an identity to the at least one virtual resource instance;
associating the at least one virtual resource instance with one virtual resource; and
translating the retrieved virtual resource into an actual resource utilized at runtime.

37. (Rejected) The method of claim 4, further comprising:
creating at least one virtual relationship instance;
assigning an identity to the at least one virtual relationship instance; and
associating the at least one virtual relationship instance with one virtual relationship.

X. EVIDENCE APPENDIX

Not applicable.

XI. RELATED PROCEEDINGS APPENDIX

Not applicable.